Sample Exam Questions

The sample exam questions that follow illustrate the relationship between the course framework and the AP Computer Science A Exam and serve as examples of the types of questions that appear on the exam. After the sample questions is a table that shows to which skill, learning objective(s), and unit each question relates. The table also provides the answers to the multiple-choice questions.

Section I: Multiple-Choice

1. Consider the following code segment.

```
int a = 5;
int b = 2;
double c = 3.0;
System.out.println(5 + a / b * c - 1);
```

What is printed when the code segment is executed?

(A) `0.6666666666666667`

(B) `9.0`

(C) `10.0`

(D) `11.5`

(E) `14.0`

2. Consider the `processWords` method. Assume that each of its two parameters is a `String` of length two or more.

```
public void processWords(String word1, String word2)
{
    String str1 = word1.substring(0, 2);
    String str2 = word2.substring(word2.length() - 1);
    String result = str2 + str1;
    System.out.println(result.indexOf(str2));
}
```

Which of the following best describes the value printed when `processWords` is called?

(A) The value `0` is always printed.

(B) The value `1` is always printed.

(C) The value `result.length() - 1` is printed.

(D) A substring containing the last character of `word2` is printed.

(E) A substring containing the last two characters of `word2` is printed.

3. Which of the following statements assigns a random integer between 25 and 60, inclusive, to `rn`?

(A) `int rn = (int) (Math.random() * 25) + 36;`

(B) `int rn = (int) (Math.random() * 25) + 60;`

(C) `int rn = (int) (Math.random() * 26) + 60;`

(D) `int rn = (int) (Math.random() * 36) + 25;`

(E) `int rn = (int) (Math.random() * 60) + 25;`

4. Vehicles are classified based on their total interior volume. The `classify` method is intended to return a vehicle classification `String` value based on total interior volume, in cubic feet, as shown in the table below.

Vehicle size class	**Total interior volume**
Minicompact	Less than 85 cubic feet
Subcompact	85 to 99 cubic feet
Compact	100 to 109 cubic feet
Mid-Size	110 to 119 cubic feet
Large	120 cubic feet or more

The `classify` method, which does not work as intended, is shown below.

```
public static String classify(int volume)
{
    String carClass = "";
    if (volume >= 120)
    {
        carClass = "Large";
    }
    else if (volume < 120)
    {
        carClass = "Mid-Size";
    }
    else if (volume < 110)
    {
        carClass = "Compact";
    }
    else if (volume < 100)
    {
        carClass = "Subcompact";
    }
    else
    {
       carClass = "Minicompact";
    }
    return carClass;
}
```

The `classify` method works as intended for some but not all values of the parameter `volume`. For which of the following values of `volume` would the correct value be returned when the `classify` method is executed?

(A) `80`

(B) `90`

(C) `105`

(D) `109`

(E) `115`

5. Which of the following best describes the value of the Boolean expression shown below?

```
a && !(b || a)
```

(A) The value is always `true`.

(B) The value is always `false`.

(C) The value is `true` when `a` has the value `false`, and is `false` otherwise.

(D) The value is `true` when `b` has the value `false`, and is `false` otherwise.

(E) The value is `true` when either `a` or `b` has the value `true`, and is `false` otherwise.

6. Consider the following code segment.

```
int val = 48;
int div = 6;
while ((val % 2 == 0) && div > 0)
{
    if (val % div == 0)
    {
       System.out.print(val + " ");
    }
    val /= 2;
    div--;
}
```

What is printed when the code segment is executed?

(A) `48 12 6`

(B) `48 12 6 3`

(C) `48 12 6 3 1`

(D) `48 24 12 6`

(E) `48 24 12 6 3`

7. Consider the following class definition.

```
public class Example
{
   private int x;
   // Constructor not shown.
}
```

Which of the following is a correct header for a method of the `Example` class that would return the value of the `private` instance variable `x` so that it can be used in a class other than `Example`?

(A) `private int getX()`

(B) `private void getX()`

(C) `public int getX()`

(D) `public void getX()`

(E) `public void getX(int x)`

8. In the following code segment, assume that the string `str` has been properly declared and initialized. The code segment is intended to print the number of strings in the array `animals` that have `str` as a substring.

```
String[] animals = {"horse", "cow", "goat", "dog",
                    "cat", "mouse"};
int count = 0;
for (int i = 0; i <= animals.length; i++)
{
    if (animals[i].indexOf(str) >= 0)
    {
       count++;
    }
}
System.out.println(count);
```

Which of the following changes should be made so the code segment works as intended?

(A) The Boolean expression in the for loop header should be changed to `i < animals.length`.

(B) The Boolean expression in the for loop header should be changed to `i < animals.length -1`.

(C) The Boolean expression in the for loop header should be changed to `i < animals[i].length`.

(D) The condition in the `if` statement should be changed to `animals[i].equals(str)`.

(E) The condition in the `if` statement should be changed to `animals[i].substring(str)`.

9. Consider an integer array, `nums`, which has been declared and initialized with one or more integer values. Which of the following code segments updates `nums` so that each element contains the square of its original value?

I.

```
int k = 0;
while (k < nums.length)
{
   nums[k] = nums[k] * nums[k];
}
```

II.

```
for (int k = 0; k < nums.length; k++)
{
   nums[k] = nums[k] * nums[k];
}
```

III.

```
for (int n : nums)
{
   n = n * n;
}
```

(A) II only

(B) I and II only

(C) I and III only

(D) II and III only

(E) I, II, and III

10. Consider the following code segment.

```
ArrayList<String> numbers = new ArrayList<String>();
numbers.add("one");
numbers.add("two");
numbers.add(0, "three");
numbers.set(2, "four");
numbers.add("five");
numbers.remove(1);
```

Which of the following represents the contents of `numbers` after the code segment has been executed?

(A) `["one", "four", "five"]`

(B) `["three", "two", "five"]`

(C) `["three", "four", "two"]`

(D) `["three", "four", "five"]`

(E) `["one", "two", "three", "four", "five"]`

11. Consider the following method, which is intended to return a list containing the elements of the parameter `myList` with all even elements removed.

```
public static ArrayList<Integer> removeEvens
   (ArrayList<Integer> myList)
{
   for (int i = 0; i < myList.size(); i++)
   {
         if (myList.get(i) % 2 == 0)
         {
             myList.remove(i);
         }
   }
   return myList;
}
```

Which of the following best explains why the code segment does not work as intended?

(A) The code segment causes an `IndexOutOfBoundsException` for all lists because of an incorrect Boolean expression in the for loop.

(B) The code segment causes an `IndexOutOfBoundsException` for lists with at least one even element because the indexes of all subsequent elements change by one when a list element is removed.

(C) The code segment returns a list with fewer elements than intended because it fails to consider the last element of `myList`.

(D) The code segment removes the wrong elements of `myList` because the condition in the `if` statement to test whether an element is even is incorrect.

(E) The code segment skips some elements of `myList` because the indexes of all subsequent elements change by one when a list element is removed.

12. Consider the following code segment.

```
int[][] points = {{11, 12, 13, 14, 15},
                  {21, 22, 23, 24, 25},
                  {31, 32, 33, 34, 35},
                  {41, 42, 43, 44, 45}};
for (int row = 0; row < points.length; row++)
{
   for (int col = points[0].length - 1;
        col >= row; col--)
   {
      System.out.print(points[row][col] + " ");
   }
   System.out.println();
}
```

What is printed when this code segment is executed?

(A)

```
15 14
25 24 23
35 34 33 32
45 44 43 42 41
```

(B)

```
15 14 13 12
25 24 23
35 34
45
```

(C)

```
11 12 13 14 15
21 22 23 24
31 32 33
41 42
```

(D)

```
15 14 13 12 11
25 24 23 22
35 34 33
45 44
```

(E)

```
15 14 13 12 11
25 24 23 22 21
35 34 33 32 31
45 44 43 42 41
```

13. Consider the following code segment.

```
int[][] arr = {{1,  2,  3,  4},
               {5,  6,  7,  8},
               {9, 10, 11, 12}};
int sum = 0;
for (int[] x : arr)
{
   for (int y = 0; y < x.length - 1; y++)
   {
      sum += x[y];
   }
}
```

What is the value of `sum` as a result of executing the code segment?

(A) `36`

(B) `54`

(C) `63`

(D) `68`

(E) `78`

14. Consider the following class definitions.

```
public class Thing1
{
   public void calc(int n)
   {
     n *= 3;
    System.out.print(n);
   }
}

public class Thing2 extends Thing1
{
   public void calc(int n)
   {
     n += 2;
     super.calc(n);
     System.out.print(n);
   }
}
```

The following code segment appears in a class other than `Thing1` or `Thing2`.

```
Thing1 t = new Thing2();
t.calc(2);
```

What is printed as a result of executing the code segment?

(A) `4`

(B) `6`

(C) `68`

(D) `124`

(E) `1212`

15. Consider the following two methods, which are intended to return the same values when they are called with the same positive integer parameter `n`.

```
public static int mystery1(int n)
{
   if (n > 1)
   {
     return 5 + mystery1(n - 1);
   }
   else
   {
     return 1;
   }
}
```

```
public static int mystery2(int n)
{
   int total = 0;
   int x = 1;
   while (x < n)
   {
     total += 5;
     x++;
   }
   return total;
}
```

Which, if any, of the following changes to `mystery2` is required so that the two methods work as intended?

(A) The variable `total` should be initialized to `1`.

(B) The variable `x` should be initialized to `0`.

(C) The condition in the `while` loop header should be `x < n - 1`.

(D) The condition in the `while` loop header should be `x <= n`.

(E) No change is required; the methods, as currently written, return the same values when they are called with the same positive integer parameter `n`.

Section II: Free-Response

The following are examples of the kinds of free-response questions found on the exam. Note that on the actual AP Exam, there will be four free-response questions.

Methods and Control Structures (Free-Response Question 1 on the AP Exam)

This question involves the use of *check digits,* which can be used to help detect if an error has occurred when a number is entered or transmitted electronically. An algorithm for computing a check digit, based on the digits of a number, is provided in part (a).

The `CheckDigit` class is shown below. You will write two methods of the `CheckDigit` class.

```
public class CheckDigit
{
    /** Returns the check digit for num, as described in part (a).
     *  Precondition: The number of digits in num is between one and
     *  six, inclusive.
     *            num >= 0
     */
     public static int getCheck(int num)
     {
         /* to be implemented in part (a) */
     }

    /** Returns true if numWithCheckDigit is valid, or false
     *  otherwise, as described in part (b).
     *  Precondition: The number of digits in numWithCheckDigit
     *  is between two and seven, inclusive.
```